Determining the Winner of a Dodgson Election is Hard*

Michael Fellows[1], Bart M. P. Jansen[2], Daniel Lokshtanov[3], Frances A. Rosamond[1], and Saket Saurabh[4]

1 **Parameterized Complexity Research Unit, University of Newcastle Callaghan, NSW Australia**
`{Michael.Fellows|Frances.Rosamond}@newcastle.edu.au`

2 **Department of Information and Computing Sciences, Utrecht University P.O. Box 80.089, 3508 TB, Utrecht, The Netherlands**
`bart@cs.uu.nl`

3 **Department of Informatics, University of Bergen N-5020 Bergen, Norway**
`daniello@ii.uib.no`

4 **Institute of Mathematical Sciences CIT Campus, Taramani, 600 113 Chennai, India**
`saket@imsc.res.in`

Abstract

Computing the *Dodgson Score* of a candidate in an election is a hard computational problem, which has been analyzed using classical and parameterized analysis. In this paper we resolve two open problems regarding the parameterized complexity of Dodgson Score. We show that Dodgson Score parameterized by the target score value k does not have a polynomial kernel unless the polynomial hierarchy collapses to the third level; this complements a result of Fellows, Rosamond and Slinko who obtain a non-trivial kernel of exponential size for a generalization of this problem. We also prove that Dodgson Score parameterized by the number n of votes is hard for $W[1]$.

1998 ACM Subject Classification F.2.2.

Keywords and phrases Dodgson Score, Parameterized Complexity, Kernelization Lower Bounds

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2010.459

1 Introduction

Complexity issues play an important role in the relatively new area of computational social choice, especially in the area of election systems, which has applications in finance and economics (agreement on the winner of an auction), internet search engines (agreement on the order of web pages presented), web mining (consensus is the notion of "public opinion"), mechanism design (agreement by participants in large networks involving autonomous software agents), and computational biology (finding consensus in feature selection), among many others [1, 14, 15]. The involvement of increasingly larger numbers of participants and the increasing sophistication of the information objects of debate, have made election systems a vital area of computer science research.

* Bart Jansen was supported by the Netherlands Organisation for Scientific Research (NWO), project "KERNELS: Combinatorial Analysis of Data Reduction".

IARCS Int'l Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2010).
Editors: Kamal Lodaya, Meena Mahajan; pp. 459–468
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

In this paper we study the hard election problem DODGSON SCORE. We consider an election in which we allow each voter to specify a complete preference ranking of the candidates: each vote is a strict total order on the set of candidates, and a vote in an election with three candidates could be represented as $a < b < c$ stating that candidate a is least preferred and c is most preferred. Given the votes that were cast in an election, we can compare the relative ranking of two candidates a, b as follows: candidate a beats candidate b in pairwise comparison if a is ranked above b more often than below b. A candidate who beats every other candidate in pairwise comparison is said to be a *Condorcet winner*. If such a winner exists then it must be unique, and it wins the election. But unfortunately a Condorcet winner may not always exist, as is shown by the following election with three candidates and three voters: $a < b < c, b < c < a, c < a < b$. This situation has a cyclic preference structure: candidate a beats b, candidate b beats c and c beats a (in pairwise comparison), so there is no candidate who beats all others. In 1876 the mathematician Charles Dodgson formulated a rule that defines the winner of an election even if there is no Condorcet winner [11, 6]. The idea is to measure how close a candidate is to being a Condorcet winner; the candidate who is closest then wins the election. This can be formalized as follows. The *Dodgson score* of a candidate c in an election, is defined to be the minimum number of swaps of adjacent candidates in the voter's preference orders that have to be made to ensure that c becomes a Condorcet winner. The candidates that have the minimum Dodgson score are the winners of the election. Dodgson's rule is not the only voting scheme resulting from Condorcet's criterion; similar schemes have been suggested by Young and Kemeny [22].

Unfortunately, DODGSON SCORE, YOUNG SCORE and KEMENY SCORE and many other election problems are NP-hard or worse, and finding an "approximate" winner of an election is hard [10] and usually not appropriate. Thus, election problems are well-suited for parameterized analysis because it offers an exact result, taking advantage of natural parameters to the problems, such as the number of votes that were cast, the number of candidates, or the score of a candidate.

1.1 Earlier Work

Bartholdi et al. initiated the study of the complexity of the Dodgson voting scheme in 1989 [2], when they showed that determining the winner of a Dodgson election is NP-hard. They also proved that computing the Dodgson or Kemeny score of a given candidate is NP-complete. The complexity of the winner problem for Dodgson elections was later established exactly; Hemaspaandra et al. [20] showed in 1997 that this problem is complete for $P_{||}^{NP}$ ("parallel access to NP").

McCabe-Dansted [21] was the first to investigate DODGSON SCORE using the framework of parameterized complexity, and observed that the ILP formulation of the problem from Bartholdi et al. [2] implies fixed-parameter tractability for the parameterization by the number m of candidates in the election. The parameterization by the target score k was first studied in 2007, when Fellows and Rosamond showed that k-DODGSON SCORE is in FPT. The group of Betzler et al. [4, 5] independently reached the same conclusion and obtained a dynamic programming algorithm with running time $O(2^k \cdot nk + nm)$ where n is the number of votes and m the number of candidates. Fellows, Rosamond and Slinko [16] considered a generalization of Dodgson's rule where each possible preference ranking specifies a cost for every swap that can be made; a candidate wins the election if the minimum total cost of making that candidate a Condorcet winner is not higher than the minimum cost of making any other candidate a Condorcet winner. They obtained a kernel of exponential size $O(e^{O(k^2)})$ for this k-GENERALIZED DODGSON SCORE problem.

The election problems Kemeny Score and Young Score have also been studied from the parameterized perspective. The Young Score problem is $W[2]$-complete when parameterized by the target score, and the same holds when using the dual of this parameter [4, 5]. The Kemeny Score problem admits several natural parameterizations that lead to fixed-parameter tractability. Results have been found for parameters 'number of votes', 'average Kendall-Tau distance', 'maximum range (or maximum Kendall-Tau distance)' and combined parameters of 'number of votes and average KT-distance', and 'number of votes and maximum KT-distance' [3].

1.2 Our Results

The parameterized analysis of Dodgson Score by Betzler et al. [5] left two open problems unanswered: 1) does k-Dodgson Score admit a polynomial kernel when parameterized by the target score, and 2) is the problem fixed-parameter tractable when parameterized by the number of votes? We answer both questions in this paper. We use the framework developed by Bodlaender et al. [7] in combination with a theorem by Fortnow and Santhanam [18] to prove that there is no polynomial kernel for k-Dodgson Score unless the polynomial hierarchy collapses to the third level (denoted as $PH = \Sigma_3^p$), and further [9]. Our second result is a non-trivial reduction establishing that k-Dodgson Score parameterized by the number of votes is hard for $W[1]$.

2 Preliminaries

In this section we formalize some notions that were introduced in Section 1. An election is a tuple (V, C) where V is a multiset of votes, and C is a set of candidates. A vote $v \in V$ is a preference list on the candidates, i.e. a strict total ordering. For candidates $a, b \in C$ the value $n_{a,b}$ counts the number of votes in V that rank a above b. A Condorcet winner is a candidate $x \in C$ such that $n_{x,y} > n_{y,x}$ for all $y \in C \setminus \{x\}$. To swap candidate x upwards in a vote $v \in V$ means to exchange the positions of x and the candidate immediately above it in the ranking; an upward swap operation is undefined if x is already the most preferred candidate in the vote. For example, if $x < z < w < y$ is a vote, then swapping x upwards once results in the vote $z < x < w < y$. We say that the candidate x *gains a vote* on candidate z through this swap, since this swap increases $n_{x,z}$ by one and decreases $n_{z,x}$ by one. The *Dodgson score* of a candidate $x \in C$ is the minimum number of swaps needed to make x a Condorcet winner. It is not hard to verify that if x can be made a Condorcet winner by k swaps, then this can also be done by k swaps that only move candidate x upwards. Consult [21, Lemma 4.0.5] for a formal proof of this claim. Therefore we may also define the Dodgson score as the minimum number of *upwards* swaps of x that are required to make x a Condorcet winner.

The theory of parameterized complexity [13] offers a toolkit for the theoretical analysis of the structure of NP-hard problems. A parameterized decision problem is a language $L \subseteq \Sigma^* \times \mathbb{N}$, where an instance (x, k) is composed of the classical input x and the parameter value k that describes some property of x. A parameterized problem L is in the class (strongly uniform) FPT (for Fixed-Parameter Tractable) if there is an algorithm that decides L in $f(k)p(|x|)$ time, where p is a polynomial and f is a computable function. A *kernelization algorithm* (kernel) [23] is a mapping that transforms an instance $(x, k) \in \Sigma^* \times \mathbb{N}$ in $p(|x| + k)$ time for some polynomial p, into an equivalent instance (x', k') such that $(x, k) \in L \Leftrightarrow (x', k') \in L$ and such that $|x'|, k' \leq f(k)$ for some computable function f. The function f is called the size of the kernel. Recent developments in the theory of kernelization have yielded tools to show that certain problems are unlikely to have kernels of polynomial size [7]. The Dodgson Score problem is formally defined as follows:

> Dodgson Score
> **Instance:** A set C of candidates, a distinguished candidate $x \in C$, a multiset V of votes and a positive integer k.
> **Question:** Can x be made a Condorcet winner by making at most k swaps between adjacent candidates?

We consider two different parameterizations in this work. When the problem is parameterized by the number of allowed swaps k then we will refer to it as k-Dodgson Score; the other variant considers a bounded number of votes $n := |V|$ which we call n-Dodgson Score.

3 Kernelization Lower Bound for k-Dodgson Score

In this section we prove that k-Dodgson Score does not have a polynomial kernel unless $PH = \Sigma_3^p$. To prove this result we need some notions related to parameterized reducibility.

▶ **Definition 1** ([8]). Let P and Q be parameterized problems. We say that P is polynomial parameter reducible to Q, written $P \leq_{Ptp} Q$, if there exists a polynomial time computable function $g : \Sigma^* \times \mathbb{N} \to \Sigma^* \times \mathbb{N}$ and a polynomial p, such that for all $(x, k) \in \Sigma^* \times \mathbb{N}$ (a) $(x, k) \in P \Leftrightarrow (x', k') = g(x, k) \in Q$ and (b) $k' \leq p(k)$. The function g is called *polynomial parameter transformation.*

▶ **Theorem 2** ([8]). *Let P and Q be parameterized problems and $\tilde{P}$ and $\tilde{Q}$ be the unparameterized versions of P and Q respectively. Suppose that $\tilde{P}$ is NP-hard and $\tilde{Q}$ is in NP. Furthermore if there is a polynomial parameter transformation from P to Q, then if Q has a polynomial kernel then P also has a polynomial kernel.*

We use the following problem as the starting point for our transformation:

> Small Universe Set Cover
> **Instance:** A set family $\mathcal{F} \subseteq 2^U$ of subsets of a finite universe U and a positive integer $k \leq |\mathcal{F}|$.
> **Question:** Is there a subfamily $\mathcal{F}' \subseteq \mathcal{F}$ with $|\mathcal{F}'| \leq k$ such that $\cup_{S \in \mathcal{F}'} S = U$?
> **Parameter:** The value $k + |U|$.

Small Universe Set Cover is a parameterized version of the NP-complete Set Cover problem [19, SP5]. We need the following incompressibility result for this problem [12, Theorem 2]:

▶ **Theorem 3.** *The problem* Small Universe Set Cover *parameterized by $k + |U|$ does not admit a polynomial kernel unless $PH = \Sigma_3^p$.*

The transformation that we shall use to prove that k-Dodgson Score does not have a polynomial kernel (unless $PH = \Sigma_3^p$) is similar in spirit to the reduction from Exact Cover by 3-sets which was originally used to show that Dodgson Score is NP-complete [2]. Let $(U, \mathcal{F}, k)$ be an instance of Small Universe Set Cover. We show how to construct an equivalent instance (V, C, x, k') of k-Dodgson Score with $k' := k(|U| + 1)$ in polynomial time. Since the problem can be solved in polynomial time when $|\mathcal{F}| < 3$, we may assume without loss of generality that the set family $\mathcal{F}$ contains at least 3 sets.

The set of candidates is composed of several parts. We create one candidate for each element u in the universe U; we will use an element $u \in U$ to refer both to the corresponding candidate and to the element of the finite universe, since the meaning will be clear from the context. We also take one candidate x to use as the distinguished candidate for whom the Dodgson score must be computed, one candidate y that will encode the fact that we must cover the universe with exactly k subsets, and finally we use three sets of dummy candidates D_0, D_1 and D_2 that are needed for padding. These dummy candidates will ensure that we can make the distance between x and y in the total orders sufficiently large, i.e. that it takes a lot of swaps for x to gain a vote on y. We want to ensure that x beats all the dummy candidates in pairwise comparison in the initial situation, to ensure that the dummies do not interfere with the encoding of the set cover instance. Our three sets of dummies D_0, D_1 and D_2 each contain $|U| + 1$ candidates. If we want to use some $d \leq |U| + 1$ dummy candidates that rank above x in the i-th vote that we create, then we use d candidates from the set $D_{i \bmod 3}$; the other dummies are ranked below x. Since x beats every dummy candidate in at least two out of three votes, this ensures that x will beat all dummy candidates in pairwise comparison if we use at least 5 votes. Using this scheme we will from now on write D^j to denote a set of $j \leq |U| + 1$ dummy candidates that can be used in the vote we are constructing.

The set V of votes is built out of two parts, each containing $|\mathcal{F}|$ votes. Since $|\mathcal{F}| \geq 3$ this will ensure that we create at least 6 votes. Using the terminology of [2] we create a set of *swing* votes corresponding to elements of $\mathcal{F}$, and a set of *equalizing* votes that create the proper initial conditions. We introduce an abbreviation to write down total orders: if $C' \subseteq C$ is a set of candidates, then by writing a total order $a < C' < b$ we mean a total order in which all candidates of C' are ranked below b and above a. The relative ranking of the candidates in C' among each other is not important. We now define the two parts of the vote set.

Swing votes. For every $S \in \mathcal{F}$ we make a vote $(\ldots < x < S < D^{|U|-|S|} < y)$. All candidates that are not explicitly mentioned in the construction are ranked below x in arbitrary order. The set S in this vote represents the candidates corresponding to the universe elements in $S \subseteq U$.

The swing votes correspond to the sets in the family $\mathcal{F}$. Observe that it takes exactly $|U|+1$ switches for x to gain a swing vote on y. The name “swing vote" comes from the fact that if x can become a Condorcet winner in $k(|U| + 1)$ switches, then all those switches must be made in swing votes.

Equalizing votes. The goal of the set of equalizing votes is to create initial conditions in which x must gain k votes on candidate y, and one vote on every candidate corresponding to some $u \in U$ in order to become the Condorcet winner. We construct the equalizing votes so that no switches made in them will allow x to become a winner in $k(|U| + 1)$ steps. We do not give an explicit construction for the equalizing votes; instead we present the conditions that they must satisfy. It will be easy to see that such a set of votes exists, and can be constructed in polynomial time.

1. For every candidate corresponding to an element $u \in U$, the number of *equalizing* votes in which x is ranked above u is equal to the number of *swing* votes in which x is ranked below u. This ensures that overall, every candidate u is ranked above x exactly as often as below x; hence x needs to gain one vote on every u to beat it in pairwise comparison.
2. There are $|\mathcal{F}| - k + 1$ equalizing votes in which x is ranked above y. Since x is ranked below y in all swing votes, this implies that there are $|\mathcal{F}| - k + 1$ votes in which x ranks above y, and $2|\mathcal{F}| - (|\mathcal{F}| - k + 1) = |\mathcal{F}| + k - 1$ votes in which x ranks below y. The reader may verify that this means that x needs to gain at least k votes on y to beat y in a pairwise comparison.

3. Whenever x is ranked below y, then (by inserting dummies if necessary) there are at least $|U|+1$ candidates between x and y. This ensures that at least $|U|+2$ swaps are needed for x to gain an equalizing vote on y.

This concludes the construction of the instance (V, C, x, k').

▶ **Lemma 4.** *If the instance $(U, \mathcal{F}, k)$ has a set cover of size k, then candidate x can be made a Condorcet winner in the election (V, C, x, k') by k' swaps.*

Proof. Suppose $\mathcal{F}' \subseteq \mathcal{F}$ is a set cover of size k. Every $S \in \mathcal{F}'$ corresponds to a swing vote. Consider the effect of swapping x upwards for $|U|+1$ steps in the swing votes corresponding to the elements of $\mathcal{F}'$. Since there are exactly $|U|$ candidates between x and y in every swing vote, this means that x gains these k votes on y. Since $\mathcal{F}'$ is a set cover of U it follows from the construction of the swing votes that we must have swapped x over every candidate $u \in U$ at least once. By the earlier observations this shows that after these $k' = k(|U|+1)$ swaps the candidate x must be a Condorcet winner. ◀

▶ **Lemma 5.** *If candidate x can be made a Condorcet winner in the election (V, C, x, k') by k' swaps, then instance $(U, \mathcal{F}, k)$ has a set cover of size k.*

Proof. Assume there is some series of $k(|U|+1)$ swaps that makes x a Condorcet winner. By the observations in the preliminaries we may assume that these swaps only move x upwards. Since x needs to gain k votes on y in order to become a Condorcet winner, we can conclude that at most $|U|+1$ swaps on average can be used for every vote that x gains over y. But by construction it is impossible to improve over y using fewer than $|U|+1$ swaps per vote, which shows that none of the swaps can be made in *equalizing votes* since there it takes at least $|U|+2$ swaps for x to improve over y. It follows that the swaps that make x a Condorcet winner in $k(|U|+1)$ steps must be composed of $|U|+1$ swaps in k different swing votes. Since x had to gain one vote on every candidate corresponding to $u \in U$ in order to become a Condorcet winner, we may conclude that in these k swing votes every candidate $u \in U$ was ranked above x at least once. But this shows that the sets corresponding to the k swing votes form a set cover for U of size k, which shows that U has a set cover of the requested size. ◀

It is not hard to verify that the transformation can be computed in polynomial time. The transformation is a polynomial parameter transformation because the parameter $k' = k(|U|+1)$ of the k-Dodgson Score instance is bounded by the square of the original parameter $k+|U|$. By combining Theorem 2 with Theorem 3 the existence of this polynomial parameter transformation yields the following theorem.

▶ **Theorem 6.** *k-Dodgson Score does not admit a polynomial kernel unless $PH = \Sigma_3^p$.*

4 Parameterized Hardness of n-Dodgson Score

We now consider the parameterization by the number of votes n and show that this leads to $W[1]$-hardness. We use a reduction from the following well-known problem [17].

Multi-Colored Clique
Instance: A simple undirected graph $G = (V, E)$, a positive integer k and a coloring function $c : V \to \{1, 2, \ldots, k\}$ on the vertices.
Question: Is there a clique in G that contains exactly one vertex from each color class?
Parameter: The value k.

We give a FPT-reduction from Multi-Colored Clique to n-Dodgson Score. In particular, given an instance $(G = (V, E), c, k)$ of Multi-Colored Clique we construct an instance (C', V', x', k') of n-Dodgson Score such that $|V'| = n = 4(\binom{k}{2} + k)$.

Let $V_1, \ldots, V_k$ be the color classes of G, that is, for every $v \in V_i$ we have $c(v) = i$. For every pair of distinct integers $1 \leq i < j \leq k$ we define $E_{i,j}$ to be the set of edges with one endpoint in V_i and one in V_j. We will assume without loss of generality that all color classes of G have the same number N of vertices, and that between every pair of color classes there are exactly M edges. We define the target score value k' of the Dodgson Score instance as $k' := ((N + 1)(Mk + 1) + 2)k + (5M - 3)\binom{k}{2}$. The set C' of candidates is built out of five groups.

1. We have a distinguished candidate x' for which we need to compute the Dodgson score.
2. We use $3k'$ dummy candidates, just as in the proof of Theorem 6. This allows us to use up to k' dummy candidates in each vote, while maintaining the property that the candidate x' is ranked above every dummy candidate in more than half of the votes.
3. For every color class $1 \leq i \leq N$ there are candidates a_i^p for $0 \leq p \leq N + 2$.
4. For every pair of color classes $1 \leq i < j \leq k$ there are candidates $a_{i,j}^p$ for $1 \leq p \leq M + 1$.
5. For every edge $e \in E_{i,j}$ there are candidates e_i, e'_i, e_j and e'_j.

From these definitions it is easy to verify that the number of candidates is polynomial in the size of the Multi-Colored Clique instance. We now describe the vote set. As in the proof of Theorem 6 we will distinguish between *swing votes* and *equalizing votes*. There are $2(\binom{k}{2} + k)$ votes of each type, and hence $|V'| = n = 4(\binom{k}{2} + k)$ from which it follows that the parameter n for the n-Dodgson Score instance is polynomial in the parameter k of the Multi-Colored Clique instance.

Equalizing votes. The equalizing votes create the right initial conditions for the election. We build the equalizing votes such that in the resulting election the distinguished candidate x' must gain exactly one vote on each non-dummy candidate in order to win the election. We ensure that no swaps made in an equalizing vote can allow x' to become a Condorcet winner in k' steps, by ranking k' dummy candidates immediately above x' in every equalizing vote. It is not hard to see that we do not need more equalizing votes than swing votes to encode these requirements.

Swing votes. The swing votes encode the behavior of the Multi-Colored Clique instance into the election. Every edge e gets an identification number $\mathrm{ID}(e)$ between 1 and M. Since the total number of edges is $M\binom{k}{2}$ the identification of two edges may be the same, but we ensure that for two distinct edges e^1, e^2 both in $E_{i,j}$ we always have $\mathrm{ID}(e^1) \neq \mathrm{ID}(e^2)$. Similarly we give every vertex $v \in V$ an identification number $\mathrm{ID}(v)$ between 1 and N, and we ensure that distinct vertices in the same color class have different ID's. As in the previous construction we know that only the part of the vote above x' is relevant, so we do not show the remainder. None of the described candidates are dummies, unless specified otherwise. For every pair of integers $1 \leq i < j \leq k$ we make two swing votes, $v_{i,j}^1$ and $v_{i,j}^2$ as follows.

$$v_{i,j}^1 : \qquad x < a_{i,j}^1 < \ldots < a_{i,j}^2 < \ldots < a_{i,j}^3 < (\ldots) < a_{i,j}^{M+1} \tag{1}$$

$$v_{i,j}^2 : \qquad x < a_{i,j}^{M+1} < \ldots < a_{i,j}^{M} < \ldots < a_{i,j}^{M-1} < (\ldots) < a_{i,j}^{1} \tag{2}$$

The gaps between consecutive candidates $a_{i,j}^p$ are filled as follows. For every edge $e \in E_{i,j}$ we insert e_i, e'_i, e_j and e'_j between $a_{i,j}^{\mathrm{ID}(e)}$ and $a_{i,j}^{\mathrm{ID}(e)+1}$ in $v_{i,j}^1$. Also, we insert e_i, e'_i, e_j and e'_j between $a_{i,j}^{\mathrm{ID}(e)+1}$ and $a_{i,j}^{\mathrm{ID}(e)}$ in $v_{i,j}^2$. Notice that between any consecutive $a_{i,j}^p$'s in $v_{i,j}^1$ and $v_{i,j}^2$ there are exactly 4 other candidates.

For every integer $1 \leq i \leq k$ we make two swing votes, v_i^1 and v_i^2 as follows.

$$v_i^1: \quad x < a_i^0 < \ldots < a_i^1 < \ldots < a_i^2 < \ldots < a_i^3 < (\ldots) < a_i^N \tag{3}$$

$$v_i^2: \quad x < a_i^{N+2} < \ldots < a_i^{N+1} < \ldots < a_i^N < \ldots < a_i^{N-1} < (\ldots) < a_i^2 \tag{4}$$

For every edge e with one endpoint v in V_i we add e_i between $a_i^{\mathrm{ID}(v)-1}$ and $a_i^{\mathrm{ID}(v)}$ in v_i^1 and we add e'_i between $a_i^{\mathrm{ID}(v)+2}$ and $a_i^{\mathrm{ID}(v)+1}$ in v_i^2. Having done this for every edge, we add dummy candidates between each consecutive pair of a_i^p's in v_i^1 and v_i^2 such that the total number of candidates between each consecutive pair of a_i^p's in v_i^1 and v_i^2 is exactly kM. This concludes the construction of (C', V', x', k').

▶ **Lemma 7.** *If G contains a colored k-clique, then x' can be made a winner of the election (C', V', x', k') in $k' = ((N+1)(kM+1)+2)k + (5M-3)\binom{k}{2}$ swaps.*

Proof. Let $C = c_1, c_2 \ldots c_k$ be a clique in G such that $c_i \in V_i$. For each vertex $c_i \in C$ we move x' in v_i^1 such that x beats $a_i^{\mathrm{ID}(c_i)}$. We also move x' in v_i^2 such that x beats $a_i^{\mathrm{ID}(c_i)+1}$. For each value of i this takes exactly $(N+1)(kM+1)+2$ swaps: we need 1 swap to move over a_i^0 in v_i^1 and 1 swap to move over a_i^{N+2} in v_i^2, and for all $N+1$ other candidates a_i^p we need to swap over the block of kM in front of them and over the candidates themselves, resulting in $(N+1)(kM+1)$ more swaps. Thus in total there are $((N+1)(kM+1)+2)k$ swaps in the v_i^1 and v_i^2 swing votes.

For every pair of distinct integers $1 \leq i < j \leq k$ we move x' in $v_{i,j}^1$ such that x beats $a_{i,j}^{\mathrm{ID}(c_i c_j)}$, and move x' in $v_{i,j}^2$ such that x beats $a_{i,j}^{\mathrm{ID}(c_i c_j)+1}$. The number of swaps to do this is $(5M-3)\binom{k}{2}$. Thus the total number of swaps is $((N+1)(kM+1)+2)k + (5M-3)\binom{k}{2} = k'$.

We show that x has gained a swing vote on each non-dummy candidate. It is easy to see that for every $1 \leq i \leq k$, the candidate x' has gained a swing vote on all a_i^p's and that for every $1 \leq i < j \leq k$, x' has gained a swing vote on all $a_{i,j}^p$'s. Observe that in the two swing votes $v_{i,j}^1$ and $v_{i,j}^2$, x' has gained a swing vote on all candidates e_i, e'_i, e_j and e'_j *except* for the four candidates corresponding to the edge $e[i,j] = c_i c_j$. Let these four candidates be $e_i[i,j]$, $e'_i[i,j]$, $e_j[i,j]$ and $e'_j[i,j]$ respectively. However, x' gains a swing vote on $e_i[i,j]$ in v_i^1, on $e'_i[i,j]$ in v_i^2, on $e_j[i,j]$ in v_j^1 and on $e'_j[i,j]$ in v_j^2. This concludes the proof of the lemma. ◀

▶ **Lemma 8.** *If x' can be made a winner of the election (C', V', x', k') in $k' = ((N+1)(kM+1)+2)k + (5M-3)\binom{k}{2}$ swaps, then G contains a colored k-clique.*

Proof. Observe that for any fixed i we need to perform at least $(N+1)(kM+1)+2$ swaps in v_i^1 and v_i^2 in total in order to gain a swing vote on all a_i^p's. Similarly for any fixed $1 \leq i < j \leq k$ we need to perform at least $5M-3$ swaps in $v_{i,j}^1$ and $v_{i,j}^2$ in total, in order to gain a swing vote on all $a_{i,j}^p$'s.

Thus, if strictly more than $(N+1)(kM+1)+2$ swaps are performed in v_i^1 and v_i^2 in total for some i, or if more than $5M-3$ swaps are performed in $v_{i,j}^1$ and $v_{i,j}^2$ in total for some i,j, then the total number of swaps performed must be greater than k' if the swaps make x' a Condorcet winner. So under the given assumptions for each i, exactly $(N+1)(kM+1)+2$ swaps are performed in v_i^1 and v_i^2 in total, and for each $1 \leq i < j \leq k$, exactly $5M-3$ swaps are performed in $v_{i,j}^1$ and $v_{i,j}^2$ in total.

For a fixed i, if x' has gained a swing vote for all the a_i^p's in v_i^1 and v_i^2, then there must be some $c_i \in V_i$ such that x' has been moved right over $a_i^{\mathrm{ID}(c_i)}$ in v_i^1 and right over $a_i^{\mathrm{ID}(c_i)+1}$ in v_i^2. Similarly, for every $1 \leq i < j \leq k$ there must be some $e[i,j] \in E_{i,j}$ such that x' has been moved right over $a_{i,j}^{\mathrm{ID}(e[i,j])}$ in $v_{i,j}^1$ and moved right over $a_{i,j}^{\mathrm{ID}(e[i,j])+1}$ in $v_{i,j}^2$. Notice that x' did not get any swing vote on $e_i[i,j]$, $e'_i[i,j]$, $e_j[i,j]$ and $e'_j[i,j]$. The only other places x' could have gotten a swing vote on these candidates are in v_i^1, v_i^2, v_j^1 and v_j^2 respectively. We prove that $e[i,j]$ is incident to c_i. Let v_i be the vertex incident to $e[i,j]$ in V_i. If $\mathrm{ID}(c_i) < \mathrm{ID}(v_i)$ then x' does not gain a swing vote on $e[i,j]_i$. Hence $\mathrm{ID}(c_i) \geq \mathrm{ID}(v_i)$. If, on the other hand, $\mathrm{ID}(c_i) > \mathrm{ID}(v_i)$ then x' does not gain a swing vote on $e'[i,j]_i$, and therefore we must have $\mathrm{ID}(c_i) \leq \mathrm{ID}(v_i)$. But then $v_i = c_i$ and therefore we know that for each i the vertex c_i that is selected in the votes v_i^1 and v_i^2 is an endpoint of the edge that was selected in the votes $v_{i,j}^1$ and $v_{i,j}^2$. Using $e[i,j]_j$ and $e'[i,j]_j$ one can similarly prove that $e[i,j]$ is incident to c_j. Hence $C = \{c_1, c_2, \ldots, c_k\}$ forms a clique in G. This concludes the proof of the lemma. ◀

The construction of (C', V', x', k') together with Lemmata 7 and 8 shows that there is a FPT-reduction from Multi-Colored Clique to n-Dodgson Score. Since it is well-known [17] that Multi-Colored Clique is hard for $W[1]$, we obtain the following result.

▶ **Theorem 9.** *n-Dodgson Score is hard for $W[1]$.*

5 Conclusions and Discussion

In this paper we answered two open problems with respect to the parameterized complexity of Dodgson Score. The parameterization k-Dodgson Score does not admit a polynomial kernel unless the polynomial hierarchy collapses, and n-Dodgson Score is hard for $W[1]$. The proof that k-Dodgson Score does not have a polynomial kernel unless $PH = \Sigma_3^p$ also implies that the exponential size kernel by Fellows et al. [16] for k-Generalized Dodgson Score cannot be improved to a polynomial kernel unless $PH = \Sigma_3^p$.

In a natural variant of the Dodgson Score problem we are given a set of votes over a set of candidates C, together with an integer k, and asked whether any candidate can be made a Condorcet Winner by performing at most k swaps. A simple construction extends the hardness result of Theorems 6 and 9 to this problem as well.

References

1. Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: Ranking and clustering. *J. ACM*, 55(5), 2008.
2. J. J. Bartholdi, C. A. Tovey, and M. A. Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare*, 6(2):157–165, 1989.
3. Nadja Betzler, Michael R. Fellows, Jiong Guo, Rolf Niedermeier, and Frances A. Rosamond. Fixed-parameter algorithms for Kemeny rankings. *Theor. Comput. Sci.*, 410(45):4554–4570, 2009.
4. Nadja Betzler, Jiong Guo, and Rolf Niedermeier. Parameterized computational complexity of Dodgson and Young elections. In *Proc. 11th SWAT*, pages 402–413, 2008.
5. Nadja Betzler, Jiong Guo, and Rolf Niedermeier. Parameterized computational complexity of Dodgson and Young elections. *Information and Computation*, 208(2):165–177, 2010.
6. Duncan Black, Robert Albert Newing, Iain McLean, Alistair McMillan, and Burt L. Monroe. *The theory of committees and elections*. Springer, 1998.
7. Hans L. Bodlaender, Rodney G. Downey, Michael R. Fellows, and Danny Hermelin. On problems without polynomial kernels. *J. Comput. Syst. Sci.*, 75(8):423–434, 2009.
8. Hans L. Bodlaender, Stéphan Thomassé, and Anders Yeo. Kernel bounds for disjoint cycles and disjoint paths. In *Proc. 17th ESA*, pages 635–646, 2009.
9. Jin-yi Cai, Venkatesan T. Chakaravarthy, Lane A. Hemaspaandra, and Mitsunori Ogihara. Competing provers yield improved Karp-Lipton collapse results. *Inf. Comput.*, 198(1):1–23, 2005.

10 Ioannis Caragiannis, Jason A. Covey, Michal Feldman, Christopher M. Homan, Christos Kaklamanis, Nikos Karanikolas, Ariel D. Procaccia, and Jeffrey S. Rosenschein. On the approximability of Dodgson and Young elections. In *Proc. 20th SODA*, pages 1058–1067, 2009.

11 C. L. Dodgson. *A method for taking votes on more than two issues.* Clarendon Press, Oxford, 1876.

12 Michael Dom, Daniel Lokshtanov, and Saket Saurabh. Incompressibility through colors and IDs. In *Proc. 36th ICALP*, pages 378–389, 2009.

13 Rod Downey and Michael R. Fellows. *Parameterized Complexity.* Monographs in computer science. Springer, New York, 1999.

14 Cynthia Dwork, Ravi Kumar, Moni Naor, and D. Sivakumar. Rank aggregation methods for the web. In *WWW*, pages 613–622, 2001.

15 Ronald Fagin, Ravi Kumar, and D. Sivakumar. Efficient similarity search and classification via rank aggregation. In *SIGMOD Conference*, pages 301–312, 2003.

16 Michael Fellows, Frances Rosamond, and Arkadii Slinko. Sensing God's will is fixed parameter tractable. Technical report, ResearchSpace@Auckland 561, 2008.

17 Michael R. Fellows, Danny Hermelin, Frances A. Rosamond, and Stéphane Vialette. On the parameterized complexity of multiple-interval graph problems. *Theor. Comput. Sci.*, 410(1):53–61, 2009.

18 Lance Fortnow and Rahul Santhanam. Infeasibility of instance compression and succinct PCPs for NP. In *Proc. 40th STOC*, pages 133–142, 2008.

19 Michael R. Garey and David S. Johnson. *Computers and Intractability, A Guide to the Theory of NP-Completeness.* W.H. Freeman and Company, New York, 1979.

20 Edith Hemaspaandra, Lane A. Hemaspaandra, and Jörg Rothe. Exact analysis of Dodgson elections: Lewis Carroll's 1876 voting system is complete for parallel access to NP. *J. ACM*, 44(6):806–825, 1997.

21 John C. McCabe-Dansted. Approximability and computational feasibility of Dodgson's rule. Master's thesis, University of Auckland, 2006.

22 I. McLean and A. Urken. *Classics of Social Choice.* University of Michigan Press, Ann Arbor, Michigan, 1995.

23 Rolf Niedermeier. *Invitation to Fixed-Parameter Algorithms.* Oxford University Press, 2006.